An Introduction to the HLA RTI

(also see paper DIS 96-14-103)

JSIMS Contractor Briefing

5 June 1996

Jim Calvin
jcalvin@ll.mit.edu

Richard Weatherly
weather@mitre.org

The RTI

- Provides a set of services common to multiple simulation domains
  - Federation Management
  - Declaration Management
  - Object Management
  - Ownership Management
  - Time Management
  - Management and testing probable additions
- Intended to be a “lean and mean” specification
- Some extensions beyond “lean and mean” where highly reusable

RTI Services
Federation Management

- Create/destroy a federation execution
- Join a federation
- Federation control
  - pause
  - resume
  - save
  - restore
  - query

RTI Services
Declaration Management

- Publish class/interaction/attributes
- Subscribe class/interaction/attributes
- Advisories on publish/subscribe

RTI Services
Object Management

- ID Request
- Instantiate/delete object
- Update attribute values
- Send interaction
- Reflect/receive attributes/interactions

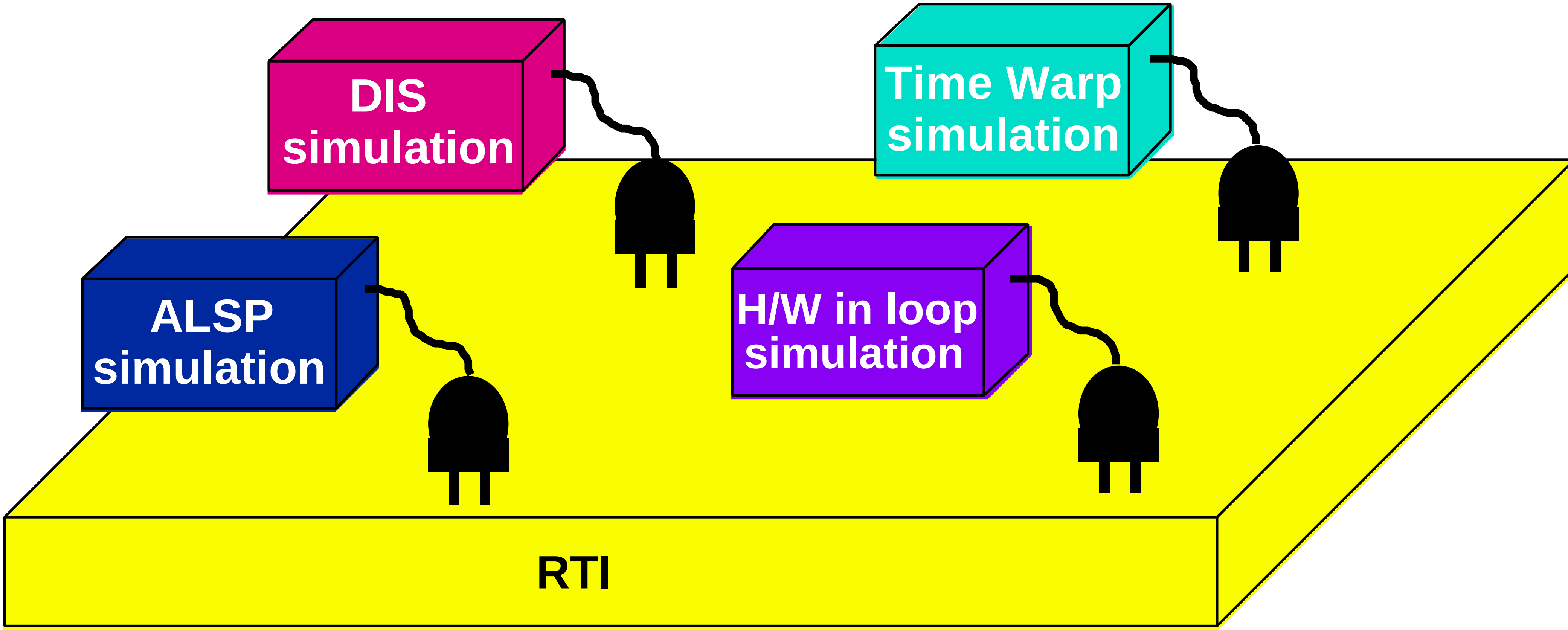
RTI Services
Ownership Management

- To be HLA compliant, the federate should be prepared to give up (or assume) the ownership of object attributes that it “knows” how to simulate
- Controls ownership of object attributes
- Brokering
  - Request divestiture
  - Divestiture notification
  - Request assumption
  - Assumption notification

RTI Services
Time Management

- The following discussion is based on
  http://www.dmso.mil/docslib/hla/TIME_MGT.DOC
  or:
  http://www.dmso.mil/docslib/hla/TIME_MGT.pdf
- And the chapter in the RTI Interface Specification
  http://www.dmso.mil/docslib/hla/ifspec04.doc
  or:
  http://www.dmso.mil/docslib/hla/IF_Spec_v0.45.pdf

Time Management Interoperability



al: A key goal of the HLA–RTI time structure is to support nteroperability among simulations with different *local* time manageme strategies in a single federation execution.

sign principle: Time Management Transparency

The *local* time management mechanism used by one simulation shou *not* be visible to other simulations in the federation.

Time Management
Interoperability

Different event ordering requirements (e.g., causal simulations and those with no event ordering requiremen

Different time advance mechanisms: coordinated, (e.g., ALSP), and independent (e.g., DIS) time advance, assuming sufficient performance of the underlying simulations and the RTI

Different time flow mechanisms: event driven, time steppe

Different synchronization mechanisms: optimistic (e.g., Time Warp) and conservative (non-rollback based) simulations

The above is a necessary, but not sufficient condition for true interoperability

Time Management Implementation Notes

- No global federation time clock (only local clocks)
  - “The federation is at time X” is an *invalid* statement
  - “The federation is at time X from the perspective of simulation i” is a valid statement.
- Assume ubiquitous, synchronized wall clock available to RTI and simulations

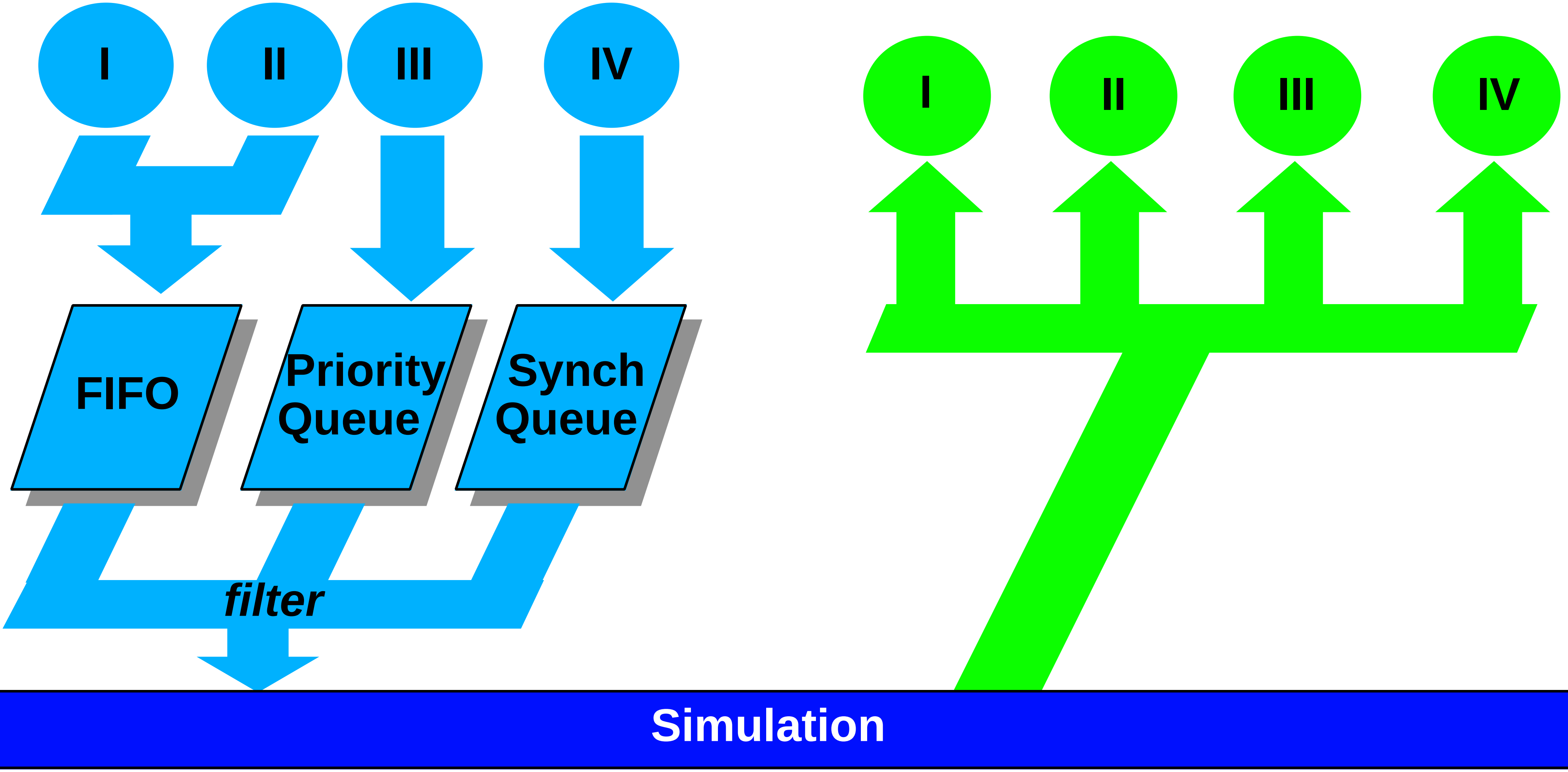
Time Management
Transportation Services

Problem statement: The HLA must support simulations with different event ordering and message delivery requirements within a single federation execution.

Approach: the HLA supports a variety of services covering

- Different reliability of delivery characteristics
  - Best effort delivery
  - Minimum rate
  - State consistent
  - Reliable delivery
- Different event ordering characteristics
  - Receive order
  - Best effort timestamp order
  - Guaranteed timestamp order

Time vs. Transportation



	Guaranteed timestamp order	Best effort timestamp order	Receive order	Comments
Reliable delivery	GTs (category IV)		RRec (category II)	ALSP like deliv
Best effort		BBts (category III)	BRec (category I)	
BE+Minimum rate		BBts (category III)	BRec (category I)	DIS -like delive
BE+State consistent		BBts (category III)	BRec (category I)	DIS /RITN-like delivery

Time Advancement Services

Time Advance Request (t)

- Requests an advance of the simulation’s logical time to t
- release all incoming BBts and GTs messages to the simulation with timestamp less than or equal to t, and all Category BRec and RRec messages

Next Event Request (t, 1 or all)

- Requests the next Category Gts event from the RTI, provided that event has a timestamp no greater than t.
- Category BRec and RRec events, and category BBts events with timestamp no greater than t released to the simulation
- A Time Advance Grant completes this request and indicates to the simulation that it has advanced its logical time to the timestamp of the single Category Gts that is delivered, if any, or to the time specified in th Next Event Request. If there are no Category Gts events with timestam less than or equal to t, and no such events will be delivered in the future a Time Advance Grant is delivered to the simulation without delivering any Category Gts events.

Request Federation Time

- Requests the current value of the local simulation clock. This clock is defined as the minimum of the real-time and logical clock of the simulation

As-Fast-As-Possible and Real-Time Federations

blem Statement: The HLA must support as-fast-as-possible federations and (scaled) real-time federations including independent and coordinated time advance simulations.

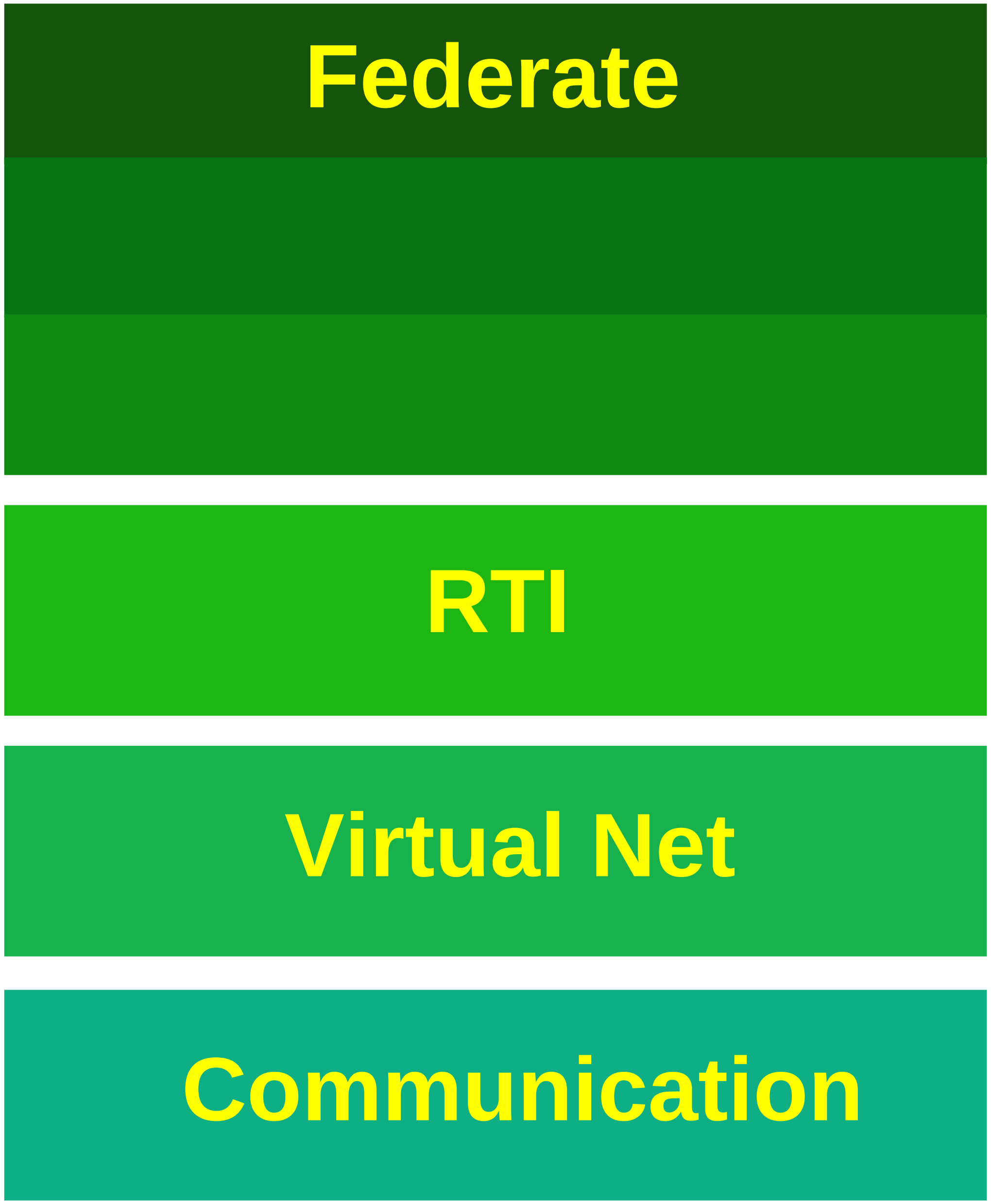
roach

- define *Logical_Time$_i$*: Time to which simulation i has progressed via **Time Advance Request** and **Next Event Request** invocations.
- define *Real_Time$_i$*: Scaled wall clock time (excluding pauses) for simulation i
- define *Local _Time$_i$:* minimum (Logical_Time$_i$, Real_Time$_i$)
- *As-fast-as-possible federation (non-real time, e.g., “batch” ALSP):* All simulatio set real-time to infinity, local clock holds “simulation time.”
- (Scaled) real-time federations:
  - *independent time advance simulation (e.g., DIS):* sets Logical_Time to in Local_Time same as Real_Time.
  - *coordinated time advance simulations (e.g., real-time ALSP)*:
    - Logical_Time$_i$ >= Real_Time$_i$: Simulation “keeping up”, paced by real-
    - Logical_Time$_i$ < Real_Time$_i$: Simulation is falling behind, progress paced by logical time advances (simulations must

RTI Services
Time Management

- Set/change logical time and rate
- Get federation time/rate
- Set/get lookahead
- Advance time
- Get next event
- Retract

A Conceptual Layering Diagram of the Prototype RTI



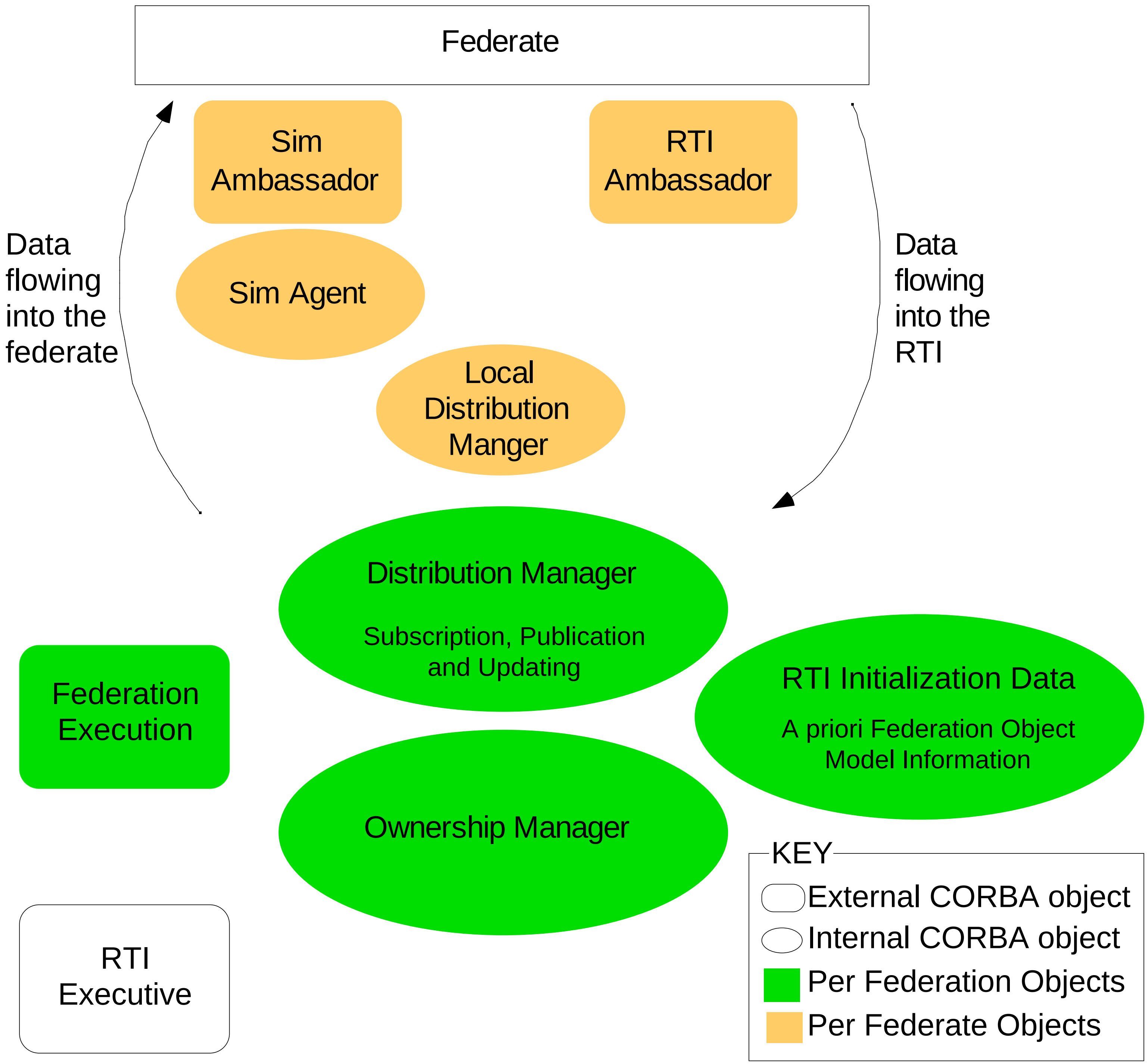
Federate

RTI

Virtual Net

Communication

Structure of the Prototype RTI



Federate

Sim Ambassador

RTI Ambassador

Sim Agent

Local Distribution Manger

Data flowing into the federate

Data flowing into the RTI

Distribution Manager

Subscription, Publication and Updating

RTI Initialization Data

A priori Federation Object Model Information

Federation Execution

Ownership Manager

RTI Executive

KEY

▭ External CORBA object

○ Internal CORBA object

🟩 Per Federation Objects

🟧 Per Federate Objects

Attributes and the RTI

- The RTI doesn’t understand the meaning of objects and their attributes, or interactions and the data associated with an interaction
- Attributes are a collection of bits to the RTI – same for interaction messages
- Attributes are based on IDL types
- The RTI attempts to provide insulation from endianess issues (multiple types of computers) via typed attributes

Current RTI Status

- Prototype RTI version 0.32 in the field
- Current version has most required functionality
  - Important additional functions to be added in the next release
- Current version does not have adequate performance for some classes of simulation
  - Work underway to address the prototype’s software architecture to improve performance
- Planning for the next release underway
  - Based on DMSO and program requirements